Richard Tolimieri
52 Ashford St.
Allston, MA 02134
July 5, 1996

Dr. Jon Sjogren
Program Manager
Communications and Statistical Research
AFOSR
110 Duncan Ave. Suite B115
AFOSR/NM Bolling AFB
DC 20332-0001

Dear Jon,

As a result of several conversations including one with Richard Orr of Atlantic Aerospace I have become concerned with the efforts of Ana Tsao, Lou Auslander and others propagandizing for the Auslander-Johnson FFT algorithm to exagerate their efforts at the expense of mine and those of my students and collaborators. I am simply interested in protecting the priority of these efforts and in no way commenting on the Auslander-Johnson efforts as they involve the integrity and influence peddling of DARPA contracts over which I have no control.

Inclosed are two papers: One published by my group in 1991 and the second recently submitted by Auslander et al in 1996 which proclaims the Auslander-Johnson FFT supported by DARPA. Both describe the exact same divide-and-conquer algorithm for finite abelian groups. The relevant pages highlighted in yellow are pages 25-26 in my work and pages 7 and 8 in Auslander's work. Note the Auslander work is equivalent to his previous efforts involving induced representations as stated on page 3. Significant extensions of this approach have been completely described in my second Springer book and in a second paper also enclosed relating the abelian group FFT to affine group actions.

These works have been submitted along with implementations to both AFOSR and DARPA from 1989, and have been mailed and hand delivered to both Auslander and Tsao. In addition, tensor product construction for the FFT matrix multiplication and the DCT along with implementations and automatic code generation have been carried out in 1989 by two of my students John Granata and Martin Rofheart who currently run their own businesses in DC aera.

Note also my paper is a tutorial survey which only claims to mathematically organize the works of others which makes it even more ironic that the new Auslander-Johnson FFT algorithm can be thought of as new five years after publication of the survey.

Sincerely,

Richard Tolimieri

Richard Tolimieri

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

0157

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
		Final Technical Report

4. TITLE AND SUBTITLE

"Weil Transform and Error Correcting Codes"

5. FUNDING NUMBERS

F49620-94-1-0012

6. AUTHOR(S)

Louis Auslander

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Graduate School and University Center/Cuny
365 Fifth Avenue
Ny, Ny 10016-4309

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of The Air Force
Air Force Office of Scientific Research
801 N. Randolph St., Arlington VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release,
distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A connection between the Weil transform and error-correcting codes is developed. This contrasts with existing efforts that are based on the Fourier transform and have led to new classes of codes well tailored for burst-error correction. A number of analytical questions must be treated in extending these analogies to the Weil case. In particular, bounds on the mean-square norm of the discrepancy between the FFT (Fast Fourier Transform) and the continuous transform should be established. This was done be consideration of a Riemann sum. The application to information theory then results by applying the Landau-Slepian "approximate dimension theorems" for functions that are only approximately band-limited. Allegations that certain results on "Fourier Transforms for Abelian Groups" which we announced as accomplishments on this grant and a related Darpa grant effort, were actually part of a well-known body of tutorial material are examined and refuted.

14. SUBJECT TERMS			15. NUMBER OF PAGES
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Review of "Radar Waveform Design and Clutter Suppression" by L. Auslander

Dr. Auslander proposes to design radar waveforms to suppress clutter in SAR images. However, he does not explain why he wants to do this! It is not clear whether he really wants to suppress clutter or sidelobes. If sidelobe suppression is the goal, then the SVA algorithm in Reference [1] is the best method (as far as I know), but it is not mentioned in Dr. Auslander's proposal.

The first two sentences of the section titled "Scientific Proposal" on Page 1 state: "Classical clutter suppression at the radar waveform level – say for Doppler – has been achieved by studying the spectrum of the clutter and designing waveforms whose Fourier transforms have little energy in the support of the spectrum of the clutter. Classical clutter suppression at the level of SAR images has been based on statistical studies of the clutter and the design of pre-whitening filters."

"Clutter" is clearly undesirable in MTI radar applications. However, in SAR applications the clutter is the signal! Saying that you want to suppress the clutter in SAR images is equivalent to saying that you want to suppress the signal of interest, unless you carefully explain what you mean by clutter suppression. Unfortunately, I could not find this explanation in the proposal.

Clutter suppression is usually considered to be a *disadvantage*, not an advantage, in SAR super-resolution algorithms. These algorithms achieve resolutions better than (1/Bandwidth) at the expense of emphasizing isolated point-like discretes over the more complicated clutter. The details of diffuse clutter features, such as vehicle tracks, are very important in SAR reconnaissance applications. Algorithms that suppress these clutter features are usually considered to be bad, not good!

There may be some benefits to suppressing diffuse clutter in imagery that will be analyzed by an automatic target recognition (ATR) algorithm, but this application is not mentioned in Dr. Auslander's proposal. (If Dr. Auslander is indeed trying to suppress diffuse clutter for an ATR algorithm, then he would need to be specific about which ATR algorithm he plans to use.)

Dr. Auslander's mathematics may lead to exotic waveforms that practical hardware cannot generate or propagate. Very high-resolution SAR images requiring very wideband waveforms are usually obtained with linear FM waveforms because it is difficult to design receiver/exciters and high-gain directive antennas with other types of very wideband waveforms. This practical issue is not addressed in the proposal.

The example provided in the proposal is based on ECG data for heart beats. Radar examples would have been much more appropriate.

In summary, my impression of the proposal is very negative.

Reference

[1] H.C. Stankwitz, R.J. Daillaire, J.R. Fienup, "Nonlinear Apodization for Sidelobe Control in SAR Imagery," *IEEE Transactions Aerospace and Electronic Systems*, January 1995.

Detailed Comments

The general problem addressed by this proposal, the construction of radar waveforms with better characteristics, is worthwhile and still merits research effort after all the years of investigation given to it.

I believe the "meat" of the proposed waveform design method for improved waveform design is contained in the bottom half of page 5 and the top half of page 6. The suggested method is intriguing and worthwhile. But many unanswered questions arise in the context of the proposed multi-resolution analysis. First, it is not entirely clear whether wavelets will be used or not. What will be the criteria used in deciding whether to use "characteristic functions" (see top of page 6) or "more sophisticated 2D wavelets?" Since there are many families of wavelets currently proposed and in vogue, what criteria will be used in selecting the most useful wavelet for the multi-resolution analysis? How will various characteristics of the wavelets chosen be separated from the signals being processed; e.g., since wavelets usually have phase responses that are not zero, how will the wavelet phases be separated from signal phase at different resolution levels?

My chief criticism is that the exciting details of demonstrating the validity and utility of the author's approach has been omitted, the reader being asked to take for granted that it will be possible to identify and articulate the details, decide upon approaches to solve the various problems, and achieve the desired goal. As a reviewer, I applaud the goal but have been disappointed to not have been given the detailed discussion to more carefully evaluate.

A second portion of the proposal, the Weil transform winding number as to new signal processing tool, should be detached from the multi-resolution discussion. This could form the basis of a separate expanded proposal that would merit evaluation on its own merits.

Arpa | 703 545 6700 |

→ 696 2287 | 696-2381

050 McCurdy

Secy Gen. Ithe. 2339.

Claire 2275

B. Lacker | 202 767 4972

767 50[illegible]16.

7756 Am C.

Lu, C., and Tolimieri, R. (1992). New algorithms for the FFT computation of symmetric and translational complex conjugate sequences. *Proc. IEEE 1992 Int. Conf. ASSP*, 23-26.
Rabiner, L. (1979). On the use of symmetry in FFT computation. *IEEE Trans. ASSSP*, **ASSSP-27**, 000-000.
Ten Eyck, L. F. (1973). Crystallographic fast Fourier transforms, *ACTA Crystallogr. Sect. A* **29**, 183-191.
Thomas, L. H. (1963). "Using a Computer to Solve Problems in Physics, Application of Digital Computers." Ginn, Waltham, MA.
Tolimieri, R., An, M., and Lu, C. (1993). "Mathematics of Multidimensional Fourier Transform Algorithms." Springer-Verlag, New York/Berlin.
Tolimieri, R., An, M., and Lu, C. (1989). "Algorithms for Discrete Fourier Transform and Convolutions." Springer-Verlag, New York/Berlin.

ADVANCES IN IMAGING AND ELECTRON PHYSICS, VOL. 93

Group Invariant Fourier Transform Algorithms[1]

R. TOLIMIERI, M. AN, Y. ABDELATIF, C. LU, G. KECHRIOTIS, and N. ANUPINDI,

AWARE Inc., One Memorial Drive, Cambridge, Massachusetts

[1] This research was supported by the Advanced Research Projects Agency of the Department Defense and was monitored by the Air Force Office of Scientific Research under contract mber F49620-91-0098. The views and conclusions contained in this document are those of e authors and should not be interpreted as necessarily representing the official policies or dorsements, either expressed or implied, of the Advanced Research Projects Agency or the .S. Government.

I. Introduction

The design of algorithms for computing the crystallographic Fourier transform is a subject in applied group theory. In previous works (An *et al.*, 1991; Tolimieri *et al.*, 1993) we exploited several elementary results in finite abelian group theory and developed the basic abstract constructs underlying the class of divide and conquer algorithms for computing the multidimensional (MD) discrete Fourier transform (DFT). This setting provides a convenient landscape for introducing a class of divide and conquer crystallographic algorithms. In An *et al.* (1991), we outlined a systematic approach for classifying three-dimensional (3D) crystallographic groups. Applications to 3D crystallography require a detailed understanding of this classification. Similar classifications exist to some extent in higher dimensions and are equally important for applications to quasicrystallography.

The theory developed in this work will operate within the abstract formulation presented in An *et al.* (1991), Tolimieri *et al.* (1993). Finite abelian groups will serve as data indexing sets. A class of *affine group* fast Fourier transform (FFT) algorithms will be introduced which fully use data invariance with respect to subgroups of the affine group of data indexing sets. The affine subgroup need not come from a crystallographic group. This approach removes dimension, transform size, and crystallographic group from algorithm design and serves to bring out fundamental algorithmic procedures rather than produce an explicit algorithm. These procedures provide tools for writing code which scales over dimension, transform size, and crystallographic group and which can be targeted to various architectures. In fact these methods apply to all 230 3D crystallographic groups and to composite transform sizes. We will show the power of these tools by way of an extensive list of implementation examples.

We distinguish three algorithm strategies. The first is based on the well-known Good–Thomas (GT) or prime factor algorithm which breaks up an FT computation into a sequence of smaller size DFT computations determined by the relatively prime factors of the initial transform sizes. In An *et al.* (1991) we developed an abstract formulation of the GT and applied it as a tool for crystallographic algorithms. Our treatment here will be brief and mostly contained in examples.

Reduced transform (RT) algorithms were considered in detail in An *et al.* (1991), Tolimieri *et al.* (1993). A simple generalization of the RT approach based on collections of subgroups will be presented, which provides a universal framework for affine group Fourier transform (FT) algorithms. In applications to 3D crystallography this class of algorithms replaces the problem of computing the FT of 3D group invariant data by that of computing in parallel the FT of a collection of 1D or 2D group-invariant data sets. The latter problem is substantially simpler and several efficient implementations are widely practiced.

A third approach, based on a generalization of Cooley–Tukey fast FT (CT FFT), will be discussed which performs generalized periodizations (Tolimieri *et al.*, 1993) with respect to affine subgroups. This method applies to abelian affine subgroup invariant data and hence to about 100 of the 230 3D crystallographic groups. A CT FFT algorithm associated to an abelian subgroup X of the affine group provides code for Y invariant data with respect to every subgroup Y of X. In applications, we choose X such that the associated CT FFT is easy to code and efficient and such that X contains a large collection of subgroups Y of interest. X itself need not be a crystallographic group. An example will be provided which shows how one code applies to 71 of the crystallographic groups.

This work is organized as follows: In Section II, we will review all the necessary group theory. Finite abelian group theory will be briefly considered as it is covered in many elementary texts. We reference Tolimieri *et al.* (1993) as it contains all the necessary results. The affine group of a finite abelian group will be defined. Constructs related to the action of affine subgroups on data indexing sets will be introduced. In Section III we define the Fourier transform of an abelian group and study its fundamental role in interchanging periodization and decimation operations (duality). The RT, CT, FFT, and GT algorithms are presented in Section IV as applications of this duality to different global decomposition strategies.

Affine group FFT algorithms based on the RT algorithm are discussed in Section VI, while those coming from the application of the affine group CT FFT are introduced in Section VIII. In Section IX, we briefly sketch a method of incorporating 1D symmetry into FFT computations, which calls on lower order existing FFT routines using the symmetry condition.

Throughout this work, we will provide many examples. These examples have been chosen to reflect both the theory and our experience and others over several years in writing code for the 3D crystallographic FT.

II. Group Theory

A. Finite Abelian Group

Denote by $\mathbf{Z}/N$ the group of integers modulo N consisting of the set

$$\{0, 1, \ldots, N-1\},$$

with addition taken modulo N. $\mathbf{Z}/N$ is a cyclic group of order N and every cyclic group of order N is isomorphic to $\mathbf{Z}/N$. For example, the

Fast Fourier Transform Algorithms for Finite Abelian Groups

L. Auslander*
Department of Mathematics
The City University of New York
New York, NY 10036

J. R. Johnson*,†
Department of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104

R. W. Johnson*,‡
Department of Computer Science
St. Cloud State University
St. Cloud, MN 56301
Phone: (612)654-9873 Fax: (612)654-9912
Email: rwj@eeyore.stcloud.msus.edu

26 July 1995

Abstract[1]— This paper presents a divide and conquer algorithm to compute the Fourier transform of a finite Abelian group using the Fourier transform of an arbitrary subgroup and the Fourier transform of the corresponding quotient group. The construction uses an arbitrary choice of coset representatives for the quotient group. Different choices of coset representatives lead to different data flows in the algorithm and different twiddle factors.

The derivation of the algorithm generalizes the derivation of the FFT presented by Cooley and Tukey. Moreover, it can be used to obtain explicit algorithms for multidimensional Fourier transforms. The algorithm presented in this paper generalizes all of the known Cooley-Tukey type algorithms for multidimensional Fourier transforms and provides new algorithms with alternative data flow patterns. Some preliminary experiments suggest that in hierarchical memory computers these algorithms are more efficient than the standard "row-column" approach.

SP EDICS SP 4.1.3 — *Multidimensional Signal Processing*
Theory, Algorithms, and Systems - Transforms

---

*Supported in part by Advanced Research Projects Agency ARPA Order No. 6674, monitored by AFOSR under contract No. F49620-89-C-0020.

†Supported in part by NSF grant CCR-9211016.

‡Supported in part by Advanced Research Projects Agency ARPA Order No. 7898, monitored by NIST under contract No. 60NANB1D1151.

[1]Permission to publish this abstract separately is granted.

96-C-6009
Martin-Marietta

1 Introduction

In 1965, Cooley and Tukey [11] presented a divide and conquer algorithm, called the Fast Fourier Transform (FFT), for computing the Fourier transform of a sequence of n points. The FFT has a long and interesting history [10].

Cooley and Tukey considered the problem of computing

$$X(j) = \sum_{k=0}^{N-1} A(k)W^{jk},\ j = 0,1,\ldots,N-1,\ W = e^{2\pi i/N} \tag{1}$$

given a complex function $A(k)$.

Their algorithm is obtained by viewing the functions X and A as functions of two variables and rearranging the sum in Equation 1 as a nested sum. Since the material in this paper is closely modeled on the ideas in the paper by Cooley and Tukey, we reproduce their derivation.

> To derive the algorithm, suppose N is composite, i.e., $N = r_1 r_2$. Then let the indices in (1) be expressed
>
> $$\begin{aligned} j &= j_1 r_1 + j_0, j_0 = 0,1,\ldots,r_1 - 1,\ j_1 = 0,1,\ldots,r_2 - 1, \\ k &= k_1 r_2 + k_0, k_0 = 0,1,\ldots,r_2 - 1,\ k_1 = 0,1,\ldots,r_1 - 1. \end{aligned} \tag{2}$$
>
> Then, one can write
>
> $$X(j_1, j_0) = \sum_{k_0}\sum_{k_1} A(k_1, k_0)W^{jk_1r_2}W^{jk_0}. \tag{3}$$
>
> Since
>
> $$W^{jk_1r_2} = W^{j_0k_1r_2}, \tag{4}$$
>
> the inner sum, over k_1, depends only on j_0 and k_0 and can be defined as a new array,
>
> $$A_1(j_0, k_0) = \sum_{k_1} A(k_1, k_0)W^{j_0k_1r_2}. \tag{5}$$
>
> The new result can then be written
>
> $$X(j_1, j_0) = \sum_{k_0} A_1(j_0, k_0)W^{(j_1r_1+j_0)k_0}. \tag{6}$$

Alternative algorithms have been presented when r_1 and r_2 are relatively prime [13]. The approach of Cooley and Tukey has been used to obtain multidimensional Fourier Transform Algorithms. The "vector radix" algorithm

was presented in [15, 23], and a more general multidimensional algorithm was independently presented in [4] and [22].

Recently the authors have generalized the Cooley-Tukey algorithm to apply to an arbitrary finite Abelian group [1]. This algorithm computes the Fourier Transform of a finite Abelian group using the Fourier transform of an arbitrary subgroup and the Fourier transform of the quotient group obtained by moding out by the chosen subgroup. The construction uses an arbitrary choice of coset representatives for the quotient group. Different choices of coset representatives lead to different data flow in the algorithm and different twiddle factors. These results generalize all of the known Cooley-Tukey type algorithms, including those mentioned above, and provides new algorithms with alternative data flow.

The results in [1] were obtained using the theory of induced representations; similar in spirit to recent work on algorithms for the computation of Fourier transforms of non-Abelian groups [8, 9, 12, 24]. In this paper an alternative combinatorial proof is presented. This proof generalizes the original proof used by Cooley and Tukey. The benefit of this approach is that it provides an explicit formula that can be used for constructing multidimensional Fourier Transform algorithms. A meta-algorithm is provided that, given a presentation for an Abelian group, a presentation for a subgroup, a set of coset representatives for the quotient group, and a set of coset representatives for a related quotient group in the dual group, constructs a matrix factorization of the Fourier transform of A. This matrix factorization provides an algorithm for computing the given multidimensional Fourier transform.

In Section 2 the Fourier transform of a finite Abelian group is defined. A proof of the main result is presented in Section 3. Section 4 shows how to apply the results about Abelian groups to multidimensional Fourier transforms, and Section 5 presents some concrete examples. In particular, it is shown how to obtain the existing algorithms as special cases of the general algorithm. In addition a class of examples are presented that could not have been computed with existing algorithms. Finally, Section 6 discusses the consequences of the results in this paper to the implementation of high performance multi-dimensional Fourier transform algorithms.

2 The Fourier Transform of a Finite Abelian Group

Let A be a finite Abelian group whose order is denoted by $|A|$ (the group operation will be denoted by $+$). The set of complex valued functions on A with inner product

$$(f,g) = \frac{1}{|A|}\sum_{a\in A} f(a)\overline{g(a)}$$

form an inner product space denoted by $L^2(A)$.

A non-zero function $\chi \in L^2(A)$ such that

$$\chi(a_1 + a_2) = \chi(a_1)\chi(a_2) \text{ for } a_1, a_2 \in A,$$

is called a character. If $a \in A$ is of order n, then $\chi(a)^n = 1$, and therefore $|\chi(a)| = 1$ and $\chi(-a) = \overline{\chi(a)}$.

If χ_1 and χ_2 are characters then we can define $(\chi_1+\chi_2)(a) = \chi_1(a)\chi_2(a)$. Under this operation the set of characters form a group, called the dual or character group, which we will denote by $\hat{A}$. It is well known that $\hat{A}$ is isomorphic to A (see Section 4 and [14]).

Suppose that $\chi \not\equiv 1$, then there exists $a_1 \in A$ such that $\chi(a_1) \neq 1$. Since,

$$\sum_{a\in A}\chi(a) = \sum_{a\in A}\chi(a + a_1) = \chi(a_1)\sum_{a\in A}\chi(a),$$

$(1-\chi(a_1))\sum_{a\in A}\chi(a) = 0$, and since $\chi(a_1) \neq 1$, $\sum_{a\in A}\chi(a) = 0$. The following property easily follows from this calculation.

Lemma 2.1 *Let* $\chi_1, \chi_2 \in \hat{A}$.

$$(\chi_1, \chi_2) = \begin{cases} 1 & \text{if } \chi_1 = \chi_2 \\ 0 & \text{if } \chi_1 \neq \chi_2 \end{cases}$$

This lemma implies that $\hat{A}$ is an orthonormal basis for $L^2(A)$. Hence, an arbitrary function, $f \in L^2(A)$ can be written uniquely as

$$f(x) = \sum_{\chi\in\hat{A}} \hat{f}(\chi)\chi(x), \text{ where } \hat{f}(\chi) = (f,\chi). \tag{7}$$

This is the Fourier series expansion of the function f. The coefficients in this expansion are called the Fourier coefficients and are obtained from the Fourier transform of f.

Definition 2.1 (Fourier Transform)
The Fourier transform of A, $F(A) : L^2(A) \to L^2(\hat{A})$ is defined by

$$F(A)(f)(\chi) = \hat{f}(\chi) = \frac{1}{|A|} \sum_{a \in A} f(a)\overline{\chi(a)} = (f, \chi).$$

3 A Divide and Conquer Algorithm

Let $\langle , \rangle : A \times \hat{A} \to \mathbf{C}$ be the bilinear pairing of A with its dual $\hat{A}$ defined by

$$\langle a, \hat{a} \rangle = \hat{a}(a).$$

Let $B < A$ be a subgroup of A. Corresponding to B we can form a subgroup, $B^\perp$, of $\hat{A}$ equal to $\{\hat{a} \in \hat{A} \mid \langle b, \hat{a} \rangle = 1, \text{for } b \in B\}$, the characters that are perpendicular, with respect to $\langle , \rangle$, to the subgroup B.

Since $\hat{a} \in B^\perp$ is equal to 1 on B, we can identify it with a character on $C = A/B$, by setting $\hat{a}(a+B) = \hat{a}(a)$. Likewise given a character, $\hat{c}$, of A/B, we can obtain a character in $B^\perp$, by composing the projection $A \to A/B$ with $\hat{c}$. Therefore $B^\perp$ is isomorphic to $\hat{C}$.

A character, $\hat{a} \in \hat{A}$, restricted to the elements of B is a character of $\hat{B}$ denoted by $\hat{a}|_B$. The restriction map, $\hat{a} \mapsto \hat{a}|_B$ is a homomorphism from $\hat{A} \to \hat{B}$ with kernel $B^\perp$. Since $|B^\perp| = |A/B|$ and $|A| = |\hat{A}|$, $|\hat{A}/B^\perp| = |\hat{B}|$ and the restriction map is onto; therefore

$$\hat{A}/B^\perp \cong \hat{B}.$$

As a result of this isomorphism there are $|B^\perp| = |C|$ characters in $\hat{A}$ that restrict to each character in $\hat{B}$. Moreover, these are the characters in the cosets of $\hat{A}/B^\perp$.

Let $C = A/B$ and $\hat{B} = \hat{A}/B^\perp$. Let $\xi : C \to A$ and $\hat{\eta} : \hat{B} \to \hat{A}$ be a choices of coset representatives. The following theorem shows how to compute $F(A)$ using $|C|$ copies of $F(B)$, $|B|$ copies of $F(C)$ and $|A|$ complex multiplications. The complex multiplications depend on the choices of coset representatives ξ and $\hat{\eta}$.

To simplify the notation used in the following theorem and proof we will remove the normalization constant $1/|A|$ and conjugation in the definition of the Fourier transform and compute

$$F(A)(\hat{a}) = \sum_{a \in A} f(a)\langle a, \hat{a} \rangle.$$

There is no harm in doing this since the constants and conjugation can easily be reinserted.

Theorem 3.1 *Let $B < A$, $C = A/B$, $\hat{C} = B^{\perp}$, $\hat{B} = \hat{A}/\hat{C}$, $\xi : C \to A$ be a choice of coset representatives for A/B, and $\hat{\eta} : \hat{B} \to \hat{A}$ be a choice of coset representatives for $\hat{A}/\hat{C}$. Then*

$$\hat{f}(\hat{a}) = \sum_{c \in C} \langle c, \hat{c} \rangle \left(\langle \xi(c), \hat{\eta}(\hat{b}) \rangle \sum_{b \in B} f_{\xi(c)}(b) \langle b, \hat{b} \rangle \right).$$

where $f_{\xi(c)}(b) = f(b + \xi(c))$ and $\hat{a} \in \hat{A} = \hat{\eta}(\hat{b}) + \hat{c}$, with $\hat{b} \in \hat{B}$ and $\hat{c} \in \hat{C}$.

Proof. Using the coset decompositions of A/B and $\hat{A}/\hat{C}$ the indexing sets can be written as $A = B \times \xi(C)$ and $\hat{A} = \hat{C} \times \hat{\eta}(\hat{B})$, and

$$\hat{f}(\hat{a}) = \sum_{a \in A} f(a) \langle a, \hat{a} \rangle \tag{8}$$

$$= \sum_{c \in C} \sum_{b \in B} f_{\xi(c)}(b) \langle b + \xi(c), \hat{c} + \hat{\eta}(\hat{b}) \rangle. \tag{9}$$

Using the bilinearity of $<,>$ this is equal to

$$\sum_{c \in C} \sum_{b \in B} f_{\xi(c)}(b) \ \langle b, \hat{c} \rangle \ \langle \xi(c), \hat{c} \rangle \ \langle b, \hat{\eta}(\hat{b}) \rangle \ \langle \xi(c), \hat{\eta}(\hat{b}) \rangle, \tag{10}$$

Since $\hat{c} \in B^{\perp}$, $< b, \hat{c} >= 1$ and Equation 10 is equal to

$$\sum_{c \in C} \sum_{b \in B} f_{\xi(c)}(b) \langle \xi(c), \hat{c} \rangle \langle b, \hat{\eta}(\hat{b}) \rangle \langle \xi(c), \hat{\eta}(\hat{b}) \rangle. \tag{11}$$

Furthermore, since $\langle \xi(c), c \rangle$ and $\langle \xi(c), \hat{\eta}(\hat{b}) \rangle$ do not depend on the inner summation index b, this is equal to the nested sum

$$\sum_{c \in C} \langle \xi(c), \hat{c} \rangle \left(\langle \xi(c), \hat{\eta}(\hat{b}) \rangle \sum_{b \in B} f_{\xi(c)}(b) \langle b, \hat{\eta}(\hat{b}) \rangle \right). \tag{12}$$

Finally, since $\langle \xi(c), \hat{c} \rangle$ and $\langle b, \hat{\eta}(\hat{b}) \rangle$ are independent of the choice of coset representatives $\xi(c)$ and $\hat{\eta}(\hat{b})$ this completes the proof. ∎

This theorem is the basis for a divide and conquer algorithm for computing $F(A)$, which we now describe.

1. Compute $\hat{f}_{\xi(c)} = F(B)f_{\xi(c)}$ for $c \in C$.

2. Compute $g_{\hat{\eta}(\hat{b})}$ for $\hat{b} \in \hat{B}$, where $g_{\hat{\eta}(\hat{b})}(c) = \langle \xi(c), \hat{\eta}(\hat{b}) \rangle \hat{f}_{\xi(c)}(\hat{b})$ for $c \in C$.

3. Compute $\hat{g}_{\hat{\eta}(\hat{b})} = F(C)g_{\hat{\eta}(\hat{b})}$ for $\hat{b} \in \hat{B}$.

Observe that $\hat{f}_{\xi(c)} \in L^2(\hat{B})$, $g_{\hat{\eta}(\hat{b})} \in L^2(C)$, $\hat{g}_{\hat{\eta}(\hat{b})} \in L^2(\hat{C})$, and by Theorem 3.1 $\hat{g}_{\hat{\eta}(\hat{b})}(\hat{c}) = \hat{f}(\hat{\eta}(\hat{b}) + \hat{c})$. Since any $\hat{a}$ can be written as $\hat{\eta}(\hat{b}) + \hat{c}$ for some $\hat{b} \in \hat{B}$ and $\hat{c} \in \hat{C}$, all of the values of the function $\hat{f} = F(A)f$ have been computed.

To implement this algorithm the elements of A and $\hat{A}$ must be ordered. Ordering the elements of A and $\hat{A}$ fixes the representations of the functions f and $\hat{f}$, introducing data permutations or index computations when accessing the values of $f_{\xi(c)}$ and $g_{\hat{\eta}(\hat{b})}$, and storing the values of $\hat{g}_{\hat{\eta}(\hat{b})}$.

If the elements of A are ordered then f can be viewed as a vector whose indices are the elements of A. Step (1) of the above algorithm first creates $|C|$ functions of B denoted by $f_{\xi(c)}$. If the elements of B are ordered then the functions $f_{\xi(c)}$ can be represented by subvectors obtained from f. The subvectors $f_{\xi(c)}$ can be ordered using the order of the coset representatives $\xi(C)$. Therefore the first part of Step (1) corresponds to a permutation of the vector f. This permutation is determined by the order of the elements of A, the subgroup B, and the coset representatives $\xi(C)$. It corresponds to permuting the indexing set A to $\xi(C) \times B$.

After the vectors $f_{\xi(c)}$ are gathered, $\hat{f}_{\xi(c)} = F(B)f_{\xi(c)}$ is computed for each $c \in C$. Step (2) then creates $|\hat{B}| = |B|$ functions of C. For each $\hat{b} \in \hat{B}$ the function $g_{\hat{\eta}(\hat{b})}$ defines a function of C. If the functions $\hat{f}_{\xi(c)}$ are stored as vectors the computation of $g_{\hat{\eta}(\hat{b})}$ requires a stride permutation followed by a diagonal multiplication (called the "twiddle factor"). The stride permutation gathers the elements of each vector $\hat{f}_{\xi(c)}$ indexed by $\hat{b}$. The twiddle factor multiplies $\hat{f}_{\xi(c)}(\hat{b})$ by $\langle \xi(c), \hat{\eta}(\hat{b}) \rangle$.

Finally, Step (3) performs $|B|$ computations of the Fourier transform on C, $\hat{g}_{\hat{\eta}(\hat{b})} = F(C)g_{\hat{\eta}(\hat{b})}$. As mentioned above, the resulting functions combine to give all of the values of $\hat{f}$; however, the values are not necessarily in the order specified for $\hat{A}$. Instead, the functions are indexed by $\hat{\eta}(\hat{B}) \times \hat{C}$, and a permutation similar to the one in Step (1) is required so that the resulting function is ordered corresponding to the order of $\hat{A}$.

Finite Fourier Transform Approximation and Riemann Sum Approximation for Functions that Decay in Time and Frequency

J. Litwin

May 1995

Abstract

For functions with finite time and frequency energy moments, we find upper bounds for the error of finite Fourier transform approximation to the Fourier transform. The error can be measured as the maximum error over all of the points of the FFT, or using a discrete L^2 distance, or using a continuous L^2 distance.

Using the machinery developed, we also find an upper bound for the error of approximating the L^2 norm of a function by a Riemann sum. From this result, an upper bound is also derived for the error of approximating the L^2 inner product, as well as the error of approximating the integral of an L^1 function, by a Riemann sum.

As an application of the L^2 norm approximation theorem, we prove an analog of the Landau-Pollak-Slepian approximate dimension theorems for a certain set of functions that is approximately time-and-bandlimited for large duration N and bandwidth M. This set can be approximately parameterized with NM parameters, with the error approaching zero as NM approaches infinity.

Contents

0 Introduction

One of the most important operations in the signal processing industry is taking the Fourier transform of a function. With digital signal processing, the finite Fourier transform (FFT) is used instead. A question of great practical importance immediately arises: How good an approximation is the FFT?

The Fourier transform at a particular frequency is nothing more than an integral. And, the FFT at a particular frequency is a special case of numerical integration using Riemann sum approximation. It would not be surprising if the theory of Riemann sum approximation allows us to derive an error bound for the FFT. In fact, we will do just the opposite. We will use an error bound for the FFT to derive an error bound for more general Riemann sum approximation.

To obtain any results, it is clear that the class of functions must be restricted somehow. For example, if there are no restrictions on the oscillations, knowing the function at a finite set of points tells us nothing about the function anywhere else. We will consider functions that decay sufficiently fast in time and frequency (in the sense that will be described below). For this class of functions, we will derive an error bound for FFT approximation to the Fourier transform and an error bound for more general Riemann approximation.

0.1 Background

Previous Results

All previous error bounds that I am aware of come out of the observation that we can rearrange the terms of the Poisson summation formula and the formula becomes the desired error bound. It is not clear who first discovered this. See Butzer-Stens [BS] and Briggs-Henson [BH, Chapter 6].

Assume that f satisfies whatever conditions are necessary so that the Poisson summation formula is valid. Consider the case of approximating the Fourier transform, denoted $\hat{f}(\gamma)$, by the FFT at $\gamma = 0$. That is, we will approximate

$$\hat{f}(0) = \int f(t)dt \quad \text{by} \quad \frac{1}{M} \sum_{k=-NM/2}^{NM/2-1} f(k/M).$$

N and M will be assumed throughout this introduction to be even positive numbers. N is the timelength of the approximation and M is the sampling rate.

Note 1: By focusing on what seems to be just the special case of FFT approxi-

mation where $\gamma = 0$, we are actually considering the general case of Riemann sum approximation.

Note 2: For the case of $\gamma \neq 0$, we can apply the following method to $fe^{-2\pi it\gamma}$. In fact, by the same method, we can obtain an upper bound for the maximum error over all the points upon which the FFT is defined. (See Section 5.1.)

One form of the Poisson summation formula is

$$\sum_{k\in\mathbf{Z}} \hat{f}(kM) = \frac{1}{M}\sum_{k\in\mathbf{Z}} f(k/M).$$

A rearrangement of the terms yields the following error bound:

$$\left|\hat{f}(0) - \frac{1}{M}\sum_{k=-\frac{1}{2}NM}^{\frac{1}{2}NM-1} f(k/M)\right| \leq \left|\frac{1}{M}\Big(\sum_{k<-\frac{1}{2}NM} + \sum_{k>\frac{1}{2}NM-1}\Big) f(k/M)\right| + \left|\sum_{k\neq 0} \hat{f}(kM)\right|.$$

The error bound can be interpreted as follows. Discretization results in two types of error. Truncation error comes from the fact that the FFT only utilizes function values over a finite duration. Aliasing error comes from the fact that the FFT is limited to a finite sampling rate. The first term of the error bound is primarily caused by the truncation error. The second term is primarily caused by the aliasing error.

The form of the error bound provides practical information about how the error can be reduced. Let's say the first term of the error bound is much greater than the second. Then increasing N will be much more effective in reducing the error than increasing M.

At this point, we can impose various conditions on f to obtain many different theorems. One possibility is as follows. Let j be an integer greater than 1. Let's assume that $f^{(j-1)}$ is absolutely continuous (which implies that $f^{(j)}$ exists almost everywhere) and that $f^{(k)} \in \mathbf{L}^1(\mathbf{R})$ for all $k \leq j$.

Then, using integration by parts j times,

$$\hat{f}(\gamma) = \int f(t)e^{-2\pi it\gamma}dt = \left(\frac{1}{2\pi i\gamma}\right)^j \int f^{(j)}(t)e^{-2\pi it\gamma}dt.$$

We will show in the next paragraph that the boundary terms from all of the integrations by parts are zero.

Consider the following two typical terms involved in one of the the above j integrations by parts:

$$\lim_{A,B\to\infty} \left(\frac{1}{2\pi i\gamma}\right)^k \int_{-A}^{B} f^{(k)}(t)e^{-2\pi it\gamma}dt \quad \text{and} \quad \lim_{A,B\to\infty} \left(\frac{1}{2\pi i\gamma}\right)^{k+1} \int_{-A}^{B} f^{(k+1)}(t)e^{-2\pi it\gamma}dt.$$

Because of our $\mathbf{L}^1$ assumption, these limits exist for any fixed γ and all $0 \leq k \leq j-1$. Therefore, the boundary term from each of the integrations by parts must also approach a limit as A, B approaches infinity. The boundary term is

$$\lim_{A,B\to\infty}\left(-(\frac{1}{2\pi i\gamma})^{k+1}f^{(k)}(B)e^{-2\pi iB\gamma}+(\frac{1}{2\pi i\gamma})^{k+1}f^{(k)}(-A)e^{2\pi iA\gamma}\right).$$

Because of the exponential factors, the only way that this limit can exist is if $\lim_{|t|\to\infty} f^{(k)}(t) = 0$. Therefore, the boundary term also equals zero.

From the expression for $\hat{f}$ in terms of $f^{(j)}$, we see that $\hat{f}$ is $O(|\gamma|^{-j})$. Now, let's estimate the second error term in the rearrangement of the Poisson summation formula. The decay of $\hat{f}$ tells us that the second error term is bounded by a constant times $\sum(kM)^{-j}$, which is $O(M^{-j})$.

We can also make the same assumptions for $\hat{f}$ that we did for f. Assume that $\hat{f}^{(j-1)}$ is absolutely continuous and that $\hat{f}^{(k)} \in \mathbf{L}^1(\mathbf{R})$ for all $k \leq j$. By similar reasoning, we see that f is $O(|t|^{-j})$. Therefore, the first error term is bounded by a constant times $\int_{N/2}^{\infty} t^{-j}dt$, which is $O(N^{-(j-1)})$.

$\mathbf{L}^2$ Approach

The prior results show that if the derivatives of f and $\hat{f}$ are absolutely continuous and in $\mathbf{L}^1(\mathbf{R})$, then we can obtain error bounds for both FFT approximation and Riemann sum approximation. What happens if we replace the hypothesis that the derivatives are in $\mathbf{L}^1$ with the hypothesis that the derivatives are in $\mathbf{L}^2$? That is the question that we will address.

Note that we are weakening the hypothesis (on the first $j-1$ derivatives). We saw above that if $f^{(k)}$ is absolutely continuous and $f^{(k)}, f^{(k+1)} \in \mathbf{L}^1$, then $\lim_{|t|\to\infty} f^{(k)}(t) = 0$. This implies that $|f^{(k)}(t)|^2 < |f^{(k)}(t)|$ for sufficiently large t which implies that $f^{(k)} \in \mathbf{L}^2$. Since our hypothesis is weaker, we should expect weaker results.

There is a simpler way to describe the conditions in our hypothesis. The assumption that $f^{(j-1)}$ is absolutely continuous and that $f^{(k)} \in \mathbf{L}^2(\mathbf{R})$ for all $k \leq j$ is equivalent to the assumption that the j^{th} frequency moment is finite. This moment is defined by

$$D_j^2 = \int \gamma^{2j}|\hat{f}|^2 d\gamma.$$

This result is essentially the content of Lemmas 1.3 and 1.4.

The dual of this result is also true. The assumption that $\hat{f}^{(j-1)}$ is absolutely continuous and that $\hat{f}^{(k)} \in \mathbf{L}^2(\mathbf{R})$ for all $k \leq j$ is equivalent to the assumption

that the j^{th} time moment is finite. This moment is defined by

$$C_j^2 = \int t^{2j}|f|^2 dt.$$

This equivalence may make our results more practical than the results obtained with the $\mathbf{L}^1$ hypothesis. With the $\mathbf{L}^1$ hypothesis, the error bound depends on the $\mathbf{L}^1$ norms of the j^{th} derivatives. This information is usually difficult to calculate. Our approach uses energy moments, which may be easier to calculate.

0.2 Notation and Definitions

The version of the Fourier transform that we use is

$$\hat{f}(\gamma) = \lim_{A\to\infty} (\text{in } \mathbf{L}^2) \int_{-A}^{A} f(t)e^{-2\pi i t\gamma} dt.$$

The limit refers to convergence in $\mathbf{L}^2(\mathbf{R})$.

$||f||$ denotes the norm of f in $\mathbf{L}^2(\mathbf{R})$. The $\mathbf{L}^2([-N/2, N/2])$ norm is denoted by $||f||_{[-N/2,N/2]}$.

The following norm is described in more detail in Section 2.2. Let $\frac{1}{M}\mathbf{Z}_N$ be the set of points $\{k/M\}_{k=-\frac{1}{2}NM,\ldots,\frac{1}{2}NM-1}$. This notation is supposed to remind the reader of the set of points $\{k/M\}$ modulo N. The $\mathbf{L}^2$ norm for functions defined on $\frac{1}{M}\mathbf{Z}_N$ is defined to be

$$||f||^2_{D(M,N)} = \frac{1}{M} \sum_{k=-\frac{1}{2}NM}^{\frac{1}{2}NM-1} |f(k/M)|^2.$$

Our version of the FFT is an operator from $\mathbf{L}^2(\frac{1}{M}\mathbf{Z}_N)$ onto $\mathbf{L}^2(\frac{1}{N}\mathbf{Z}_M)$ defined by

$$\tilde{f}(n/N) = \frac{1}{M} \sum_{k=-\frac{1}{2}NM}^{\frac{1}{2}NM-1} f(k/M)e^{-2\pi ikn/NM}, \quad n = -\frac{1}{2}NM, \ldots, \frac{1}{2}NM - 1.$$

The FFT of f will be denoted by $\tilde{f}$.

Note: The nonstandard scaling in the definitions of the FFT and the discrete norm are used to make the FFT is a unitary operator. With these definitions,

$$||\tilde{f}||_{D(N,M)} = ||f||_{D(M,N)}.$$

(See Section 2.2 for proof.)

The FFT can easily be extended to all of $\mathbf{R}$ so that it is a periodic function with period M. The *extended FFT*, denoted by $f^\natural$, is

$$f^\natural(\gamma) = \frac{1}{M} \sum_{k=-\frac{1}{2}NM}^{\frac{1}{2}NM-1} f(k/M) e^{-2\pi i k\gamma/M}.$$

Note: The extended FFT is also a unitary operator. By Parseval's equality for periodic functions,

$$||f^\natural||_{[-M/2,M/2]} = ||f||_{D(M,N)}.$$

The periodization operator P_N is defined by

$$(P_N f)(t) = \sum_{k\in\mathbf{Z}} f(t+kN), \quad f \in \mathbf{L}^2(\mathbf{R}) \text{ and decays sufficiently fast.}$$

See Section 2.2 for a more precise definition.

$\overline{x}$ means the the complex conjugate of x.

$\lfloor x \rfloor$ means the greatest integer less than or equal to x.

0.3 Executive Summary

FFT Approximation

From now on, throughout this paper, all functions are assumed to be in $\mathbf{L}^2(\mathbf{R})$, unless stated otherwise. We also assume that f and $\hat{f}$ are continuous. (In fact, most of our results depend on the existence of time and frequency energy moments, which is a stronger condition than the continuity of f and $\hat{f}$.)

Because of our assumptions, both the Fourier transform and the FFT exist. We would like to measure the error in approximating the Fourier transform by the FFT.

The error in the FFT approximation can be measured in many ways. One way is to measure the maximum difference over all of the points upon which the FFT is defined.

Since $\hat{f}$ is assumed continuous, we can evaluate $\hat{f}$ at $\frac{1}{N}\mathbf{Z}_M$. Therefore, we can also measure the error by calculating the discrete $\mathbf{L}^2$ distance, $||\hat{f} - \hat{f}||_{D(N,M)}$.

We can also measure the error by calculating the continuous $\mathbf{L}^2$ distance, $||f^{\natural} - \hat{f}||_{[-M/2,M/2]}$.

With the $\mathbf{L}^1$ hypothesis, a key step in deriving an error bound was to show that f and $\hat{f}$ decay at a sufficiently fast rate. We obtain a similar result easily in Lemma 1.5. If C_j and D_1 are finite, then

$$|f^2(t)| \leq A|t|^{-j},$$

where A is a constant. Unfortunately, this lemma doesn't give us any information about what the constant is.

However, a different approach yields an even stronger result. Define the discrete tail energy to be

$$\tilde{E}^2_{N,M} = \frac{1}{M} \sum_{|k|>\frac{1}{2}NM} |f(k/M)|^2.$$

In Lemma 2.3, we show that if C_j and D_1 are finite, then the discrete tail energy is $O(N^{-j})$, and we also have an upper bound for the constant involved.

Notice that this implies that $f^2(t)$ is $O(|t|^{-j})$. At this point, we could plug this information into the rearrangement of the Poisson summation formula as we did with the $\mathbf{L}^1$ hypothesis to obtain a bound for the maximum error. This approach results in the first FFT approximation theorem (Theorem 5.1).

By proceding somewhat differently, we can obtain bounds for the discrete and continuous $\mathbf{L}^2$ error. The following briefly describes how we derive an upper bound for the discrete $\mathbf{L}^2$ error.

Since the discrete energy in the tail is bounded, one would expect that we can find an upper bound for the discrete $\mathbf{L}^2$ distance between a function and its periodization. In fact, we prove the following periodization comparison lemma (Lemma 2.8). If C_j and D_1 are finite, where $j > 2$, then

$$||P_N f - f||_{D(M,N)} \leq K N^{-j/2},$$

where K depends only on C_j and D_1 (and inversely on N and M).

This lemma allows us to replace $\tilde{f}$ and $\hat{f}$ in the definition of the discrete $\mathbf{L}^2$ distance between the FFT and the Fourier transform with periodizations (using the triangle inequality). Of course, replacing $\tilde{f}$ with $(P_N f)^{\sim}$ introduces an error of $K_1 N^{-j/2}$ and replacing $\hat{f}$ with $P_M \hat{f}$ introduces an error of $K_2 M^{-j/2}$.

Now we must calculate $||(P_N f)^{\sim} - P_M \hat{f}||_{D(N,M)}$. For this, we use the generalized Poisson summation formula (Theorem 4.1). The generalized Poisson summation formula says that under the hypotheses of the Poisson summation formula,

$(P_N f)^{\sim}$ is equal to the sampled $P_M \hat{f}$. Therefore, in our case, where the hypotheses are met because of energy moment conditions, the norm of the difference is zero.

We have just derived the desired upper bound for the discrete $\mathbf{L}^2$ error (Theorem 5.2). If C_j and D_j are finite, where $j > 2$, then

$$||\tilde{f} - \hat{f}||_{D(N,M)} \leq K_1 N^{-j/2} + K_2 M^{-j/2}.$$

K_1 depends only on C_j and D_1 (and inversely on N and M). K_2 depends only on D_j and C_1 (and inversely on N and M).

We also obtain an upper bound for the continuous $\mathbf{L}^2$ error (Theorem 5.3, see Section 5.3 for the proof). If C_j and D_j are finite, where $j > 2$, then

$$||f^{\natural} - \hat{f}||_{[-M/2,M/2]} \leq K_1 N^{-j/2} + K_2 M^{-j}.$$

K_1 depends only on C_j and D_1 (and inversely on N and M). K_2 depends only on D_j. Notice that the second error term is better than the corresponding term in the previous result.

Riemann Sum Approximation

Since we are focusing on the $\mathbf{L}^2$ norm error, our approach does not automatically provide an error bound for more general Riemann sum approximation. However, from the third FFT approximation theorem, we derive the $\mathbf{L}^2$ norm approximation theorem (Theorem 6.1). Then, we bootstrap our way to derive error bounds for more general Riemann sum approximation.

From the third FFT approximation theorem and the triangle inequality, we have

$$\left| ||f^{\natural}||_{[-M/2,M/2]} - ||\hat{f}||_{[-M/2,M/2]} \right| \leq K_1 N^{-j/2} + K_2 M^{-j},$$

As mentioned above, the first term of this inequality, $||f^{\natural}||_{[-M/2,M/2]}$, is equal to $||f||_{D(M,N)}$ by Parseval's equality for periodic functions.

The second term, $||\hat{f}||_{[-M/2,M/2]}$, can be replaced by $||\hat{f}|| = ||f||$ at the expense of increasing K_2. This is a consequence of Chebyshev's inequality.

Making these two substitutions proves the $\mathbf{L}^2$ norm approximation theorem. If C_j and D_j are finite, where $j > 2$, then

$$\left| ||f|| - ||f||_{D(M,N)} \right| \leq K_1 N^{-j/2} + K_2 M^{-j}.$$